

```

*****
28461 Thu Jun 23 11:18:55 2016
new/usr/src/head/nss_dbdefs.h
7126 NSS_XbyY_FINI has too much logic for a function-like macro
*****
_____unchanged_portion_omitted_____

643 /* status returned by the str2ent parsing routines */
644 #define NSS_STR_PARSE_SUCCESS 0
645 #define NSS_STR_PARSE_PARSE 1
646 #define NSS_STR_PARSE_ERANGE 2

648 #define NSS_XbyY_INIT(str, res, bufp, len, func)    (\
649     (str)->buf.result = (res),                    \
650     (str)->buf.buffer = (bufp),                  \
651     (str)->buf buflen = (len),                    \
652     (str)->stayopen = 0,                          \
653     (str)->str2ent = (func),                      \
654     (str)->key2str = NULL,                        \
655     (str)->returnval = 0,                         \
656     (str)->returnlen = 0,                         \
657     (str)->h_errno = 0,                          \
658     (str)->erange = 0)

660 #define NSS_XbyY_INIT_EXT(str, res, bufp, len, func, kfunc)    (\
661     (str)->buf.result = (res),                    \
662     (str)->buf.buffer = (bufp),                  \
663     (str)->buf buflen = (len),                    \
664     (str)->stayopen = 0,                          \
665     (str)->str2ent = (func),                      \
666     (str)->key2str = (kfunc),                    \
667     (str)->returnval = 0,                         \
668     (str)->returnlen = 0,                         \
669     (str)->h_errno = 0,                          \
670     (str)->erange = 0)

672 #define NSS_XbyY_FINI(str)        _nss_XbyY_fini(str)
672 #define NSS_XbyY_FINI(str)        (\
673     (str)->returnval == 0 && (str)->erange && (errno = ERANGE), \
674     (str)->returnval)

674 #define NSS_PACKED_CRED_CHECK(buf, ruid, euid)    (\
675     ((nss_pheader_t *) (buf))->p_ruid == (ruid) && \
676     ((nss_pheader_t *) (buf))->p_euid == (euid))

678 extern void        *_nss_XbyY_fini(nss_XbyY_args_t *);
679 #endif /* ! codereview */
680 extern char        **_nss_netdb_aliases(const char *, int, char *, int);
681 extern nss_status_t nss_default_key2str(void *, size_t, nss_XbyY_args_t *,
682     const char *, int, size_t *);
683 extern nss_status_t nss_packed_arg_init(void *, size_t, nss_db_root_t *,
684     nss_db_initf_t *, int *,
685     nss_XbyY_args_t *);
686 extern nss_status_t nss_packed_context_init(void *, size_t, nss_db_root_t *,
687     nss_db_initf_t *, nss_getent_t **,
688     nss_XbyY_args_t *);
689 extern void        nss_packed_set_status(void *, size_t, nss_status_t,
690     nss_XbyY_args_t *);
691 extern nss_status_t nss_packed_getkey(void *, size_t, char **, int *,
692     nss_XbyY_args_t *);

694 /*
695 * nss_dbop_t values for searches with various keys; values for
696 * destructor/endent/setent/getent are defined in <nss_common.h>
697 */

```

```

699 /*
700 * These are part of the "Over the wire" IE app->nscd getXbyY
701 * op for well known getXbyY's.  Cannot use NSS_DBOP_X_Y directly
702 * because NSS_DBOP_next_iter is NOT an incrementing counter value
703 * it's a starting offset into an array value.
704 */

706 #define NSS_DBOP_X(x)                ((x)<<16)
707 #define NSS_DBOP_XY(x, y)           ((x)|(y))

709 #define NSS_DBOP_ALIASES             NSS_DBOP_X(1)
710 #define NSS_DBOP_AUTOMOUNT          NSS_DBOP_X(2)
711 #define NSS_DBOP_BOOTPARAMS         NSS_DBOP_X(3)
712 #define NSS_DBOP_ETHERS             NSS_DBOP_X(4)
713 #define NSS_DBOP_GROUP              NSS_DBOP_X(5)
714 #define NSS_DBOP_HOSTS              NSS_DBOP_X(6)
715 #define NSS_DBOP_IPNODES            NSS_DBOP_X(7)
716 #define NSS_DBOP_NETGROUP           NSS_DBOP_X(8)
717 #define NSS_DBOP_NETMASKS           NSS_DBOP_X(9)
718 #define NSS_DBOP_NETWORKS           NSS_DBOP_X(10)
719 #define NSS_DBOP_PASSWD             NSS_DBOP_X(11)
720 #define NSS_DBOP_PRINTERS           NSS_DBOP_X(12)
721 #define NSS_DBOP_PROJECT             NSS_DBOP_X(13)
722 #define NSS_DBOP_PROTOCOLS          NSS_DBOP_X(14)
723 #define NSS_DBOP_PUBLICKEY          NSS_DBOP_X(15)
724 #define NSS_DBOP_RPC                NSS_DBOP_X(16)
725 #define NSS_DBOP_SERVICES           NSS_DBOP_X(17)
726 #define NSS_DBOP_AUDITUSER         NSS_DBOP_X(18)
727 #define NSS_DBOP_AUTHATTR           NSS_DBOP_X(19)
728 #define NSS_DBOP_EXECATTR           NSS_DBOP_X(20)
729 #define NSS_DBOP_PROFATTR           NSS_DBOP_X(21)
730 #define NSS_DBOP_USERATTR           NSS_DBOP_X(22)

732 #define NSS_DBOP_GROUP_BYNAME        (NSS_DBOP_next_iter)
733 #define NSS_DBOP_GROUP_BYGID        (NSS_DBOP_GROUP_BYNAME + 1)
734 #define NSS_DBOP_GROUP_BYMEMBER     (NSS_DBOP_GROUP_BYGID + 1)

736 #define NSS_DBOP_PASSWD_BYNAME      (NSS_DBOP_next_iter)
737 #define NSS_DBOP_PASSWD_BYUID       (NSS_DBOP_PASSWD_BYNAME + 1)

739 /* The "compat" backend requires that PASSWD_BYNAME == SHADOW_BYNAME */
740 /* (it also requires that both use key.name to pass the username). */
741 #define NSS_DBOP_SHADOW_BYNAME      (NSS_DBOP_PASSWD_BYNAME)

743 #define NSS_DBOP_PROJECT_BYNAME      (NSS_DBOP_next_iter)
744 #define NSS_DBOP_PROJECT_BYID       (NSS_DBOP_PROJECT_BYNAME + 1)

746 #define NSS_DBOP_HOSTS_BYNAME       (NSS_DBOP_next_iter)
747 #define NSS_DBOP_HOSTS_BYADDR       (NSS_DBOP_HOSTS_BYNAME + 1)

749 #define NSS_DBOP_IPNODES_BYNAME     (NSS_DBOP_next_iter)
750 #define NSS_DBOP_IPNODES_BYADDR     (NSS_DBOP_IPNODES_BYNAME + 1)

752 /*
753 * NSS_DBOP_NAME_2ADDR
754 * NSS_DBOP_ADDR_2NAME
755 *
756 */
: are defines for ipv6 api's

758 #define NSS_DBOP_NAME_2ADDR          (NSS_DBOP_next_ipv6_iter)
759 #define NSS_DBOP_ADDR_2NAME         (NSS_DBOP_NAME_2ADDR + 1)

761 #define NSS_DBOP_RPC_BYNAME          (NSS_DBOP_next_iter)
762 #define NSS_DBOP_RPC_BYNUMBER        (NSS_DBOP_RPC_BYNAME + 1)

```

```

764 #define NSS_DBOP_NETWORKS_BYNAME      (NSS_DBOP_next_iter)
765 #define NSS_DBOP_NETWORKS_BYADDR     (NSS_DBOP_NETWORKS_BYNAME + 1)

767 #define NSS_DBOP_SERVICES_BYNAME     (NSS_DBOP_next_iter)
768 #define NSS_DBOP_SERVICES_BYPORT    (NSS_DBOP_SERVICES_BYNAME + 1)

770 #define NSS_DBOP_PROTOCOLS_BYNAME    (NSS_DBOP_next_iter)
771 #define NSS_DBOP_PROTOCOLS_BYNUMBER (NSS_DBOP_PROTOCOLS_BYNAME + 1)

773 #define NSS_DBOP_ETHERS_HOSTTON      (NSS_DBOP_next_noiter)
774 #define NSS_DBOP_ETHERS_NTOHOST     (NSS_DBOP_ETHERS_HOSTTON + 1)

776 #define NSS_DBOP_BOOTPARAMS_BYNAME   (NSS_DBOP_next_noiter)
777 #define NSS_DBOP_NETMASKS_BYNET     (NSS_DBOP_next_noiter)

779 #define NSS_DBOP_PRINTERS_BYNAME     (NSS_DBOP_next_iter)

781 /*
782  * The "real" backend for netgroup (__multi_innetgr, setnetgrent)
783  */
784 #define NSS_DBOP_NETGROUP_IN         (NSS_DBOP_next_iter)
785 #define NSS_DBOP_NETGROUP_SET       (NSS_DBOP_NETGROUP_IN + 1)

787 /*
788  * The backend for getpublickey and getsecretkey (getkeys)
789  */
790 #define NSS_DBOP_KEYS_BYNAME        (NSS_DBOP_next_iter)

792 /*
793  * The pseudo-backend for netgroup (returned by setnetgrent) doesn't have
794  * any getXXXbyYYY operations, just the usual destr/end/set/get ops,
795  * so needs no definitions here.
796  */

798 #define NSS_DBOP_ATTRDB_BYNAME       (NSS_DBOP_next_iter)

800 #define NSS_DBOP_AUDITUSER_BYNAME    NSS_DBOP_ATTRDB_BYNAME
801 #define NSS_DBOP_AUTHATTR_BYNAME    NSS_DBOP_ATTRDB_BYNAME
802 #define NSS_DBOP_EXECATTR_BYNAME    NSS_DBOP_ATTRDB_BYNAME
803 #define NSS_DBOP_EXECATTR_BYID      (NSS_DBOP_EXECATTR_BYNAME + 1)
804 #define NSS_DBOP_EXECATTR_BYNAMEID  (NSS_DBOP_EXECATTR_BYID + 1)
805 #define NSS_DBOP_PROFATTR_BYNAME    NSS_DBOP_ATTRDB_BYNAME
806 #define NSS_DBOP_USERATTR_BYNAME    NSS_DBOP_ATTRDB_BYNAME

808 #define NSS_DBOP_TSOL_TP_BYNAME      (NSS_DBOP_next_iter)
809 #define NSS_DBOP_TSOL_RH_BYADDR     (NSS_DBOP_next_iter)
810 #define NSS_DBOP_TSOL_ZC_BYNAME     (NSS_DBOP_next_iter)

812 /*
813  * Used all over in the switch code. The best home for it I can think of.
814  * Power-of-two alignments only.
815  */
816 #define ROUND_DOWN(n, align)        (((uintptr_t)n) & ~((align) - 1))
817 #define ROUND_UP(n, align)          ROUND_DOWN(((uintptr_t)n) + (align) - 1, \
818 (align))

820 #ifdef __cplusplus
821 }
822 #endif

824 #endif /* _NSS_DBDEFS_H */

```

new/usr/src/lib/libc/port/gen/nss\_common.c

1

\*\*\*\*\*

59657 Thu Jun 23 11:18:57 2016

new/usr/src/lib/libc/port/gen/nss\_common.c

7126 NSS\_XbyY\_FINI has too much logic for a function-like macro

\*\*\*\*\*

\_\_\_\_\_unchanged\_portion\_omitted\_\_\_\_\_

```
1022 static void
1023 output_loop_diag_a(
1024     int n,
1025     char *dbase,
1026     struct __nsw_lookup_v1 *lkp)
```

```
1027 {
1028     (void) fprintf(__nss_debug_file,
1029         "NSS_retry(%d): '%s': trying '%s' ... ",
1030         n, dbase, lkp->service_name);
1031     (void) fflush(__nss_debug_file);
1033 }
```

```
1035 static void
1036 output_loop_diag_b(
1037     nss_status_t res,
1038     struct __nsw_lookup_v1 *lkp)
```

```
1039 {
1040     (void) fprintf(__nss_debug_file, "result=");
1041     switch (res) {
1042     case NSS_SUCCESS:
1043         (void) fprintf(__nss_debug_file, "SUCCESS");
1044         break;
1045     case NSS_NOTFOUND:
1046         (void) fprintf(__nss_debug_file, "NOTFOUND");
1047         break;
1048     case NSS_UNAVAIL:
1049         (void) fprintf(__nss_debug_file, "UNAVAIL");
1050         break;
1051     case NSS_TRYAGAIN:
1052         (void) fprintf(__nss_debug_file, "TRYAGAIN");
1053         break;
1054     case NSS_NISSERVDNS_TRYAGAIN:
1055         (void) fprintf(__nss_debug_file, "NISSERVDNS_TRYAGAIN");
1056         break;
1057     default:
1058         (void) fprintf(__nss_debug_file, "undefined");
1059     }
1060     (void) fprintf(__nss_debug_file, ", action=");
1061     switch (lkp->actions[res]) {
1062     case __NSW_CONTINUE:
1063         (void) fprintf(__nss_debug_file, "CONTINUE");
1064         break;
1065     case __NSW_RETURN:
1066         (void) fprintf(__nss_debug_file, "RETURN");
1067         break;
1068     case __NSW_TRYAGAIN_FOREVER:
1069         (void) fprintf(__nss_debug_file, "TRYAGAIN_FOREVER");
1070         break;
1071     case __NSW_TRYAGAIN_NTIMES:
1072         (void) fprintf(__nss_debug_file, "TRYAGAIN_NTIMES (N=%d)",
1073             lkp->max_retries);
1074         break;
1075     case __NSW_TRYAGAIN_PAUSED:
1076         (void) fprintf(__nss_debug_file, "TRYAGAIN_PAUSED");
1077         break;
1078     default:
```

new/usr/src/lib/libc/port/gen/nss\_common.c

2

```
1079         (void) fprintf(__nss_debug_file, "undefined");
1080     }
1081     (void) fprintf(__nss_debug_file, "\n");
1082 }
```

\_\_\_\_\_unchanged\_portion\_omitted\_\_\_\_\_

```
2300 void *
2301 __nss_XbyY_fini(nss_XbyY_args_t *args)
2302 {
2303     if ((args->returnval == NULL) && (args->erange != 0))
2304         errno = ERANGE;
2305     return (args->returnval);
2306 }
2307 #endif /* ! codereview */
```

\*\*\*\*\*

58438 Thu Jun 23 11:18:58 2016

new/usr/src/lib/libc/port/mapfile-vers

7126 NSS\_XbyY\_FINI has too much logic for a function-like macro

\*\*\*\*\*

\_\_\_\_\_unchanged\_portion\_omitted\_\_\_\_\_

2793 # There should never be more than one SUNWprivate version.

2794 # Don't add any more. Add new private symbols to SUNWprivate\_1.1

```

2796 SYMBOL_VERSION SUNWprivate_1.1 {
2797     global:
2798         __Argv          { FLAGS = NODIRECT };
2799         cfree           { FLAGS = NODIRECT };
2800         __cswidth;
2801         __ctype_mask;
2802         __environ_lock { FLAGS = NODIRECT };
2803         __inf_read;
2804         __inf_written;
2805         __i_size;
2806         __isnanf       { TYPE = FUNCTION; FILTER = libm.so.2 };
2807         __iswrunes;
2808         __libc_threaded;
2809         __lib_version  { FLAGS = NODIRECT };
2810         __logb         { TYPE = FUNCTION; FILTER = libm.so.2 };
2811         __lone         { FLAGS = NODYNSORT };
2812         __lten         { FLAGS = NODYNSORT };
2813         __lzero        { FLAGS = NODYNSORT };
2814         __malloc_lock;
2815         __memcmp;
2816         __memcpy       { FLAGS = NODYNSORT };
2817         __memmove;
2818         __memset;
2819         __modff        { TYPE = FUNCTION; FILTER = libm.so.2 };
2820         __nan_read;
2821         __nan_written;
2822         __nextwctype;
2823         __nis_debug_bind;
2824         __nis_debug_calls;
2825         __nis_debug_file;
2826         __nis_debug_rpc;
2827         __nis_prefsrv;
2828         __nis_preftype;
2829         __nis_server;
2830         __nss_default_finders;
2831         __progname     { FLAGS = NODIRECT };
2832         __smbuf;
2833         __sp;
2834         __strdupa_str  { FLAGS = NODIRECT };
2835         __strdupa_len { FLAGS = NODIRECT };
2836         __tdb_bootstrap;
2837         __threaded;
2838         thr_probe_getfunc_addr;
2839         __trans_lower;
2840         __trans_upper;
2841         __uberdata;
2842         __xpg6         { FLAGS = NODIRECT };
2844 $if _ELF32
2845     __dladdr          { TYPE = FUNCTION; FILTER = /usr/lib/ld.so.1 };
2846     __dladdr1         { TYPE = FUNCTION; FILTER = /usr/lib/ld.so.1 };
2847     __dlclose         { TYPE = FUNCTION; FILTER = /usr/lib/ld.so.1 };
2848     __dldump          { TYPE = FUNCTION; FILTER = /usr/lib/ld.so.1 };
2849     __dlerror         { TYPE = FUNCTION; FILTER = /usr/lib/ld.so.1 };

```

```

2850     __dlinfo          { TYPE = FUNCTION; FILTER = /usr/lib/ld.so.1 };
2851     __dlmopen         { TYPE = FUNCTION; FILTER = /usr/lib/ld.so.1 };
2852     __dlopen          { TYPE = FUNCTION; FILTER = /usr/lib/ld.so.1 };
2853     __dlsym           { TYPE = FUNCTION; FILTER = /usr/lib/ld.so.1 };
2854     __ld_libc         { TYPE = FUNCTION; FILTER = /usr/lib/ld.so.1 };
2855     __sys_errlist;
2856     __sys_errs;
2857     __sys_index;
2858     __sys_nerr        { FLAGS = NODYNSORT };
2859     __sys_num_err;
2860 $elif sparcv9
2861     __dladdr          { TYPE = FUNCTION; FILTER = /usr/lib/sparcv9/ld.so.1 };
2862     __dladdr1         { TYPE = FUNCTION; FILTER = /usr/lib/sparcv9/ld.so.1 };
2863     __dlclose         { TYPE = FUNCTION; FILTER = /usr/lib/sparcv9/ld.so.1 };
2864     __dldump          { TYPE = FUNCTION; FILTER = /usr/lib/sparcv9/ld.so.1 };
2865     __dlerror         { TYPE = FUNCTION; FILTER = /usr/lib/sparcv9/ld.so.1 };
2866     __dlinfo         { TYPE = FUNCTION; FILTER = /usr/lib/sparcv9/ld.so.1 };
2867     __dlmopen        { TYPE = FUNCTION; FILTER = /usr/lib/sparcv9/ld.so.1 };
2868     __dlopen          { TYPE = FUNCTION; FILTER = /usr/lib/sparcv9/ld.so.1 };
2869     __dlsym           { TYPE = FUNCTION; FILTER = /usr/lib/sparcv9/ld.so.1 };
2870     __ld_libc         { TYPE = FUNCTION; FILTER = /usr/lib/sparcv9/ld.so.1 };
2871 $elif amd64
2872     __dladdr          { TYPE = FUNCTION; FILTER = /usr/lib/amd64/ld.so.1 };
2873     __dladdr1         { TYPE = FUNCTION; FILTER = /usr/lib/amd64/ld.so.1 };
2874     __dlamd64getunwind { TYPE = FUNCTION; FILTER = /usr/lib/amd64/ld.so.1 };
2875     __dlclose         { TYPE = FUNCTION; FILTER = /usr/lib/amd64/ld.so.1 };
2876     __dldump          { TYPE = FUNCTION; FILTER = /usr/lib/amd64/ld.so.1 };
2877     __dlerror         { TYPE = FUNCTION; FILTER = /usr/lib/amd64/ld.so.1 };
2878     __dlinfo         { TYPE = FUNCTION; FILTER = /usr/lib/amd64/ld.so.1 };
2879     __dlmopen        { TYPE = FUNCTION; FILTER = /usr/lib/amd64/ld.so.1 };
2880     __dlopen          { TYPE = FUNCTION; FILTER = /usr/lib/amd64/ld.so.1 };
2881     __dlsym           { TYPE = FUNCTION; FILTER = /usr/lib/amd64/ld.so.1 };
2882     __ld_libc         { TYPE = FUNCTION; FILTER = /usr/lib/amd64/ld.so.1 };
2883 $else
2884 $error unknown platform
2885 $endif
2887 $if _sparc
2888     __lyday_to_month;
2889     __mon_lengths;
2890     __yday_to_month;
2891 $endif
2892 $if i386
2893     __sse_hw;
2894 $endif
2896     protected:
2897     acctctl;
2898     allocids;
2899     __assert_c99;
2900     __assert_c99;
2901     __assfail;
2902     attr_count;
2903     attr_to_data_type;
2904     attr_to_name;
2905     attr_to_option;
2906     attr_to_xattr_view;
2907     __autofssys;
2908     __bufsync;
2909     __cladm;
2910     __class_quadruple;
2911     core_get_default_content;
2912     core_get_default_path;
2913     core_get_global_content;
2914     core_get_global_path;
2915     core_get_options;

```

```

2916 core_get_process_content;
2917 core_get_process_path;
2918 core_set_default_content;
2919 core_set_default_path;
2920 core_set_global_content;
2921 core_set_global_path;
2922 core_set_options;
2923 core_set_process_content;
2924 core_set_process_path;
2925 dbm_close_status;
2926 dbm_do_nextkey;
2927 dbm_setdefwrite;
2928 _D_cplx_div;
2929 _D_cplx_div_ix;
2930 _D_cplx_div_rx;
2931 _D_cplx_mul;
2932 defclose_r;
2933 defcntl;
2934 defcntl_r;
2935 defopen;
2936 defopen_r;
2937 defread;
2938 defread_r;
2939 _delete;
2940 _dgettext;
2941 _doprnt;
2942 _doscan;
2943 _errfp;
2944 _errxfp;
2945 exportfs;
2946 _F_cplx_div;
2947 _F_cplx_div_ix;
2948 _F_cplx_div_rx;
2949 _F_cplx_mul;
2950 __fgetwc_xpg5;
2951 __fgetws_xpg5;
2952 _findbuf;
2953 _findiop;
2954 __fini_daemon_priv;
2955 _finite;
2956 _forkl { FLAGS = NODYNSORT };
2957 _forkall { FLAGS = NODYNSORT };
2958 _fpclass;
2959 _fpgetmask;
2960 _fpgetround;
2961 _fpgetsticky;
2962 _fprintf;
2963 _fpsetmask;
2964 _fpsetround;
2965 _fpsetsticky;
2966 __fputwc_xpg5;
2967 __fputws_xpg5;
2968 _ftw;
2969 _gcvt;
2970 _getarg;
2971 __getcontext;
2972 _getdents;
2973 _get_exit_frame_monitor;
2974 _getfp;
2975 _getgroupsbymember;
2976 _getlogin_r;
2977 getrandom;
2978 _getsp;
2979 __gettsp;
2980 getvmusage;
2981 __getwchar_xpg5;

```

```

2982 __getwc_xpg5;
2983 gtty;
2984 __idmap_flush_kcache;
2985 __idmap_reg;
2986 __idmap_unreg;
2987 __init_daemon_priv;
2988 __init_suid_priv;
2989 _insert;
2990 inst_sync;
2991 _iswctype;
2992 klpd_create;
2993 klpd_getpath;
2994 klpd_getport;
2995 klpd_getucred;
2996 klpd_register;
2997 klpd_register_id;
2998 klpd_unregister;
2999 klpd_unregister_id;
3000 __lgrp_home_fast { FLAGS = NODYNSORT };
3001 __lgrpsys;
3002 _lltostr;
3003 _lock_clear;
3004 _lock_try;
3005 _ltzset;
3006 lwp_self;
3007 makeut;
3008 makeutx;
3009 __mbftowc;
3010 mcfiller;
3011 mnptopt;
3012 modctl;
3013 modutx;
3014 msgctl64;
3015 __multi_innetgr;
3016 __mutex_destroy { FLAGS = NODYNSORT };
3017 mutex_enter;
3018 mutex_exit;
3019 mutex_held;
3020 __mutex_init { FLAGS = NODYNSORT };
3021 __mutex_unlock { FLAGS = NODYNSORT };
3022 name_to_attr;
3023 nfs_getfh;
3024 nfssvc;
3025 _nfssys;
3026 __nis_get_environment;
3027 _nss_db_state_destr;
3028 nss_default_key2str;
3029 nss_delete;
3030 nss_endent;
3031 nss_getent;
3032 __nss_initf_group;
3033 __nss_initf_netgroup;
3034 __nss_initf_passwd;
3035 __nss_initf_shadow;
3036 nss_packed_arg_init;
3037 nss_packed_context_init;
3038 nss_packed_getkey;
3039 nss_packed_set_status;
3040 nss_search;
3041 nss_setent;
3042 __nss_XbyY_fgets;
3043 __nss_XbyY_fini;
3044 #endif /* ! codereview */
3045 __nsw_extended_action_v1;
3046 __nsw_freeconfig_v1;
3047 __nsw_getconfig_v1;

```

```

3048     __nthreads;
3049     __openattrdirat;
3050     option_to_attr;
3051     __priv_bracket;
3052     __priv_relinquish;
3053     pset_assign_forced;
3054     pset_bind_lwp;
3055     _psignal;
3056     pthread_attr_getdaemonstate_np;
3057     pthread_attr_setdaemonstate_np;
3058     _pthread_setcleanupinit;
3059     __putwchar_xpg5;
3060     __putwc_xpg5;
3061     rctlctl;
3062     rctl1list;
3063     _realbufend;
3064     _resume;
3065     _resume_ret;
3066     _rpcsys;
3067     _sbrk_grow_aligned;
3068     scrwidth;
3069     semctl64;
3070     _semctl64;
3071     set_setcontext_enforcement;
3072     _setbufend;
3073     __set_errno;
3074     setprojrctl;
3075     _setregid;
3076     _setreuid;
3077     setsigacthandler;
3078     shmctl64;
3079     _shmctl64;
3080     sigflag;
3081     _signal;
3082     _sigoff;
3083     _sigon;
3084     __so_accept;
3085     __so_bind;
3086     __sockconfig;
3087     __so_connect;
3088     __so_getpeername;
3089     __so_getsockname;
3090     __so_getsockopt;
3091     __so_listen;
3092     __so_recv;
3093     __so_recvfrom;
3094     __so_recvmsg;
3095     __so_send;
3096     __so_sendmsg;
3097     __so_sendto;
3098     __so_setsockopt;
3099     __so_shutdown;
3100     __so_socket;
3101     __so_socketpair;
3102     str2group;
3103     str2passwd;
3104     str2spwd;
3105     __strptime_dontzero;
3106     stty;
3107     syscall;
3108     __sysconfig;
3109     __systemcall;
3110     thr_continue_allmutators;
3111     _thr_continue_allmutators;
3112     thr_continue_mutator;
3113     _thr_continue_mutator;

```

```

3114     thr_getstate;
3115     _thr_getstate;
3116     thr_mutators_barrier;
3117     _thr_mutators_barrier;
3118     thr_probe_setup;
3119     _thr_schedctl;
3120     thr_setmutator;
3121     _thr_setmutator;
3122     thr_setstate;
3123     _thr_setstate;
3124     thr_sighndlrinfo;
3125     _thr_sighndlrinfo;
3126     _thr_slot_offset;
3127     thr_suspend_allmutators;
3128     _thr_suspend_allmutators;
3129     thr_suspend_mutator;
3130     _thr_suspend_mutator;
3131     thr_wait_mutator;
3132     _thr_wait_mutator;
3133     __tls_get_addr;
3134     _tmem_get_base;
3135     _tmem_get_nentries;
3136     _tmem_set_cleanup;
3137     tpool_create;
3138     tpool_dispatch;
3139     tpool_destroy;
3140     tpool_wait;
3141     tpool_suspend;
3142     tpool_suspended;
3143     tpool_resume;
3144     tpool_member;
3145     _tynname_dev;
3146     _ucred_alloc;
3147     ucred_getamask;
3148     _ucred_getamask;
3149     ucred_getasid;
3150     _ucred_getasid;
3151     ucred_getatid;
3152     _ucred_getatid;
3153     ucred_getauid;
3154     _ucred_getauid;
3155     _ulltostr;
3156     _uncached_getgrgid_r;
3157     _uncached_getgrnam_r;
3158     _uncached_getpwnam_r;
3159     _uncached_getpwuid_r;
3160     __ungetwc_xpg5;
3161     _unordered;
3162     utssys;
3163     _verrfp;
3164     _verrxfp;
3165     _vwarnfp;
3166     _vwarnxfp;
3167     _warnfp;
3168     _warnxfp;
3169     __wcsftime_xpg5;
3170     __wcstok_xpg5;
3171     wdbindf;
3172     wdchkind;
3173     wddelim;
3174     _wrtchk;
3175     _xflsbuf;
3176     _xgetwidth;
3177     zone_add_datalink;
3178     zone_boot;
3179     zone_check_datalink;

```

```

3180     zone_create;
3181     zone_destroy;
3182     zone_enter;
3183     zone_getattr;
3184     zone_get_id;
3185     zone_list;
3186     zone_list_datalink;
3187     zonept;
3188     zone_remove_datalink;
3189     zone_setattr;
3190     zone_shutdown;
3191     zone_version;

```

```
3193 $if _ELF32
```

```

3194     __divdi3;
3195     __file_set;
3196     __fprintf_c89;
3197     __fscanf_c89;
3198     __fwprintf_c89;
3199     __fwscanf_c89;
3200     __imaxabs_c89;
3201     __imaxdiv_c89;
3202     __moddi3;
3203     __printf_c89;
3204     __scanf_c89;
3205     __snprintf_c89;
3206     __sprintf_c89;
3207     __sscanf_c89;
3208     __strtoimax_c89;
3209     __strtoumax_c89;
3210     __swprintf_c89;
3211     __swscanf_c89;
3212     __uidivi3;
3213     __umoddi3;
3214     __vfprintf_c89;
3215     __vfscanf_c89;
3216     __vfwprintf_c89;
3217     __vfwscanf_c89;
3218     __vprintf_c89;
3219     __vscanf_c89;
3220     __vsnprintf_c89;
3221     __vsprintf_c89;
3222     __vsscanf_c89;
3223     __vswprintf_c89;
3224     __vswscanf_c89;
3225     __vwprintf_c89;
3226     __vwscanf_c89;
3227     __wcstoimax_c89;
3228     __wcstoumax_c89;
3229     __wprintf_c89;
3230     __wscanf_c89;

```

```
3231 $endif
```

```
3233 $if _sparc
```

```

3234     __cerror;
3235     install_utrap;
3236     __install_utrap;
3237     nop;
3238     __Q_cplx_div;
3239     __Q_cplx_div_ix;
3240     __Q_cplx_div_rx;
3241     __Q_cplx_lr_div;
3242     __Q_cplx_lr_div_ix;
3243     __Q_cplx_lr_div_rx;
3244     __Q_cplx_lr_mul;
3245     __Q_cplx_mul;

```

```

3246     __QgetRD;
3247     __xregs_clrptr;
3248 $endif

```

```
3250 $if sparc32
```

```

3251     __ashldi3;
3252     __ashrdi3;
3253     __cerror64;
3254     __cmpdi2;
3255     __floatdidf;
3256     __floatdisf;
3257     __floatundidf;
3258     __floatundisf;
3259     __lshrdi3;
3260     __muldi3;
3261     __ucmpdi2;
3262 $endif

```

```
3264 $if _x86
```

```

3265     __D_cplx_lr_div;
3266     __D_cplx_lr_div_ix;
3267     __D_cplx_lr_div_rx;
3268     __F_cplx_lr_div;
3269     __F_cplx_lr_div_ix;
3270     __F_cplx_lr_div_rx;
3271     __fltrounds;
3272     __sysi86;
3273     __sysi86;
3274     __X_cplx_div;
3275     __X_cplx_div_ix;
3276     __X_cplx_div_rx;
3277     __X_cplx_lr_div;
3278     __X_cplx_lr_div_ix;
3279     __X_cplx_lr_div_rx;
3280     __X_cplx_mul;
3281     __xgetRD;
3282     __xtol;
3283     __xtoll;
3284     __xtoul;
3285     __xtoull;
3286 $endif

```

```
3288 $if i386
```

```

3289     __divrem64;
3290     __tls_get_addr;
3291     __udivrem64;
3292 $endif

```

```

3294 # The following functions should not be exported from libc,
3295 # but /lib/libm.so.2, some older versions of the Studio
3296 # compiler/debugger components, and some ancient programs
3297 # found in /usr/dist reference them. When we no longer
3298 # care about these old and broken binary objects, these
3299 # symbols should be deleted.

```

```

3300     __brk                                { FLAGS = NODYNSORT };
3301     __cond_broadcast                      { FLAGS = NODYNSORT };
3302     __cond_init                          { FLAGS = NODYNSORT };
3303     __cond_signal                        { FLAGS = NODYNSORT };
3304     __cond_wait                          { FLAGS = NODYNSORT };
3305     __ecvt                                { FLAGS = NODYNSORT };
3306     __fcvt                                { FLAGS = NODYNSORT };
3307     __getc_unlocked                      { FLAGS = NODYNSORT };
3308     __llseek                             { FLAGS = NODYNSORT };
3309     __pthread_attr_getdetachstate       { FLAGS = NODYNSORT };
3310     __pthread_attr_getinheritsched     { FLAGS = NODYNSORT };
3311     __pthread_attr_getschedparam       { FLAGS = NODYNSORT };

```

```

3312  _pthread_attr_getschedpolicy    { FLAGS = NODYNSORT };
3313  _pthread_attr_getscope           { FLAGS = NODYNSORT };
3314  _pthread_attr_getstackaddr       { FLAGS = NODYNSORT };
3315  _pthread_attr_getstacksize       { FLAGS = NODYNSORT };
3316  _pthread_attr_init                { FLAGS = NODYNSORT };
3317  _pthread_condattr_getpshared      { FLAGS = NODYNSORT };
3318  _pthread_condattr_init           { FLAGS = NODYNSORT };
3319  _pthread_cond_init                { FLAGS = NODYNSORT };
3320  _pthread_create                   { FLAGS = NODYNSORT };
3321  _pthread_getschedparam            { FLAGS = NODYNSORT };
3322  _pthread_join                     { FLAGS = NODYNSORT };
3323  _pthread_key_create               { FLAGS = NODYNSORT };
3324  _pthread_mutexattr_getprioceiling { FLAGS = NODYNSORT };
3325  _pthread_mutexattr_getprotocol    { FLAGS = NODYNSORT };
3326  _pthread_mutexattr_getpshared     { FLAGS = NODYNSORT };
3327  _pthread_mutexattr_init           { FLAGS = NODYNSORT };
3328  _pthread_mutex_getprioceiling     { FLAGS = NODYNSORT };
3329  _pthread_mutex_init               { FLAGS = NODYNSORT };
3330  _pthread_sigmask                  { FLAGS = NODYNSORT };
3331  _rwlock_init                      { FLAGS = NODYNSORT };
3332  _rw_rdlock                         { FLAGS = NODYNSORT };
3333  _rw_unlock                         { FLAGS = NODYNSORT };
3334  _rw_wlock                         { FLAGS = NODYNSORT };
3335  _sbrk_unlocked                    { FLAGS = NODYNSORT };
3336  _select                           { FLAGS = NODYNSORT };
3337  _sema_init                         { FLAGS = NODYNSORT };
3338  _sema_post                        { FLAGS = NODYNSORT };
3339  _sema_trywait                     { FLAGS = NODYNSORT };
3340  _sema_wait                         { FLAGS = NODYNSORT };
3341  _sysfs                            { FLAGS = NODYNSORT };
3342  _thr_create                       { FLAGS = NODYNSORT };
3343  _thr_exit                          { FLAGS = NODYNSORT };
3344  _thr_getprio                       { FLAGS = NODYNSORT };
3345  _thr_getspecific                  { FLAGS = NODYNSORT };
3346  _thr_join                          { FLAGS = NODYNSORT };
3347  _thr_keycreate                    { FLAGS = NODYNSORT };
3348  _thr_kill                          { FLAGS = NODYNSORT };
3349  _thr_main                          { FLAGS = NODYNSORT };
3350  _thr_self                          { FLAGS = NODYNSORT };
3351  _thr_setspecific                  { FLAGS = NODYNSORT };
3352  _thr_sigsetmask                   { FLAGS = NODYNSORT };
3353  _thr_stksegment                   { FLAGS = NODYNSORT };
3354  _ungetc_unlocked                  { FLAGS = NODYNSORT };

3356  local:
3357  __imax_lldiv                       { FLAGS = NODYNSORT };
3358  __ti_thr_self                       { FLAGS = NODYNSORT };
3359  *;

```

```

3361  $if lf64
3362  __seekdir64                        { FLAGS = NODYNSORT };
3363  __telldir64                        { FLAGS = NODYNSORT };
3364  $endif

3366  $if _sparc
3367  __cerror                            { FLAGS = NODYNSORT };
3368  $endif

3370  $if sparc32
3371  __cerror64                          { FLAGS = NODYNSORT };
3372  $endif

3374  $if sparcv9
3375  __cleanup                            { FLAGS = NODYNSORT };
3376  $endif

```

```

3378  $if i386
3379  __syscall6                          { FLAGS = NODYNSORT };
3380  __systemcall6                       { FLAGS = NODYNSORT };
3381  $endif

3383  $if amd64
3384  __tls_get_addr                      { FLAGS = NODYNSORT };
3385  $endif
3386  };

```